

CLEAN CLAIMS AFTER *RESPONSE A*

1 1. (First Amended per A) A computer implemented method for emulating execution of legacy
2 instructions, where said legacy instructions have instruction addresses, comprising:
3 accessing blocks of said legacy instructions, said blocks having block addresses,
4 storing translations, into a translation store, for each of the legacy instructions,
5 storing translation indications, for indicating translated blocks, into an indexing table at
6 block numbers determined by said block addresses, said storing translation
7 indications using a subset of block address digits whereby block numbers in said
8 table are the same for multiple different blocks,
9 executing said translated instructions to emulate said legacy instructions,
10 where for each of the legacy instructions of a translated block having a block number in said
11 table, said storing translations step includes translating the legacy instructions into
12 one or more translated instructions for emulating the legacy instructions, and
13 if the legacy instruction is not a store instruction, going to said step of
14 executing said translated instructions,
15 if the legacy instruction is a store instruction, where the store instruction
16 stores to a particular block with a particular block number in said
17 table, checking the indications in said table for the particular block
18 number and,
19 if the indications indicate that said particular block has not
20 been translated, going to said step of executing said
21 translated instructions,
22 if the indications indicate that said particular block has been
23 translated, checking said translation store to determine
24 if legacy instruction data has been modified and if
25 modified, repeating the step of translating the legacy
26 instructions and going to said step of executing said
27 translated instructions; and otherwise, if legacy

CLEAN CLAIMS AFTER *RESPONSE A*

28 instruction data has not been modified, going to said
29 step of executing said translated instructions.

1 2. (Original) The method of Claim 1 wherein said step of storing translation indications stores
2 indications for only a subset of all the translated blocks.

1 3. (Original) The method of Claim 2 wherein said subset of all the translated blocks is stored in
2 a cache.

1 4. (Cancelled).

1 5. (First Amended per A) The method of Claim 1 wherein said block address digits are included
2 in a three digit hexadecimal address field and said subset of block address digits is the center digit.

1 6. (Original) The method of Claim 1 wherein said legacy instructions are for a legacy system
2 having a S/390 architecture.

1 7. (Original) The method of Claim 1 wherein said legacy instructions are object code instructions
2 compiled/assembled for a legacy architecture.

1 8. (Original) The method of Claim 1 wherein said legacy instructions include store instructions for
2 modifying instruction code.

1 9. (Original) The method of Claim 1 wherein said translation indications include a state field for
2 each block number indicating whether the block represented by said block number has been
3 modified.

CLEAN CLAIMS AFTER *RESPONSE A*

1 10. (First Amended per A) The method of Claim 1 wherein,
2 said subset of all the translated blocks is stored in a cache,
3 said translation indications include a state field storing a count for each block number
4 indicating whether the block represented by said block number has been modified,
5 said count in a state field is incremented each time a block represented by said block number
6 has been modified in said cache,
7 said count in a state field is decremented each time a block represented by said block number
8 has been removed from said cache,
9 said step of checking said translation store occurs only when said count is zero.

1 11. (First Amended per A) A computer implemented method for dynamic emulation of object code
2 legacy instructions, where the legacy instructions have instruction addresses determined by
3 compilation/assembly of source code and where the legacy instructions include self-modifying store
4 instructions for modifying instruction code, comprising:
5 accessing blocks of said legacy instructions, said blocks having block addresses,
6 storing translations, into a translation store, for each of the legacy instructions,
7 storing translation indications, for only a subset of all the translated blocks, into an indexing
8 table at block numbers determined by said block addresses, said storing translation
9 indications,
10 using a subset of block address digits whereby block numbers in said table
11 are the same for multiple different blocks,
12 including a state field storing a count for each block number indicating
13 whether the block represented by said block number has been
14 modified by self-modifying store instructions,
15 executing said translated instructions to emulate said legacy instructions,
16 where for each of the legacy instructions of said subset of all the translated blocks having a
17 block number in said table,

CLEAN CLAIMS AFTER *RESPONSE A*

18 said storing translations step includes translating the legacy instruction into
19 one or more translated instructions for emulating the legacy
20 instruction and storing said translated instructions in a cache,
21 if the legacy instruction is not a store instruction, going to said step of
22 executing said translated instructions,
23 if the legacy instruction is a store instruction, where the store instruction
24 stores to a particular block with a particular block number in said
25 table, checking the indications in said table for said particular block
26 number and,
27 if the indications indicate that said particular block has not
28 been translated, going to said step of executing said
29 translated instructions,
30 if the indications indicate that said particular block has been
31 translated, checking said translation store to determine
32 if legacy instruction data has been modified and if
33 modified, repeating the step of translating the legacy
34 instructions and going to said step of executing said
35 translated instructions; and otherwise, if instruction
36 data has not been modified going to said step of
37 executing said translated instructions.

1 12. (First Amended per A) The method of Claim 11 wherein said count in a state field is
2 incremented each time a block represented by said block number has been modified in said cache,
3 said count in a state field is decremented each time a block represented by said block number has
4 been removed from said cache, said step of checking said translation store occurs only when said
5 count is zero.

CLEAN CLAIMS AFTER *RESPONSE A*

1 13. (Original) The method of Claim 11 wherein said legacy code is compiled/assembled for a
2 native architecture and executes as a guest on a host architecture.

1 14. (Original) The method of Claim 13 wherein the native architecture employs CISC instructions
2 and the host architecture employs RISC instructions.

1 15. (First Amended per A) A computer system for emulating execution of legacy instructions,
2 where said legacy instructions have instruction addresses, comprising:

3 a group access unit for accessing blocks of said legacy instructions, said blocks having block
4 addresses,

5 a translator for translating the legacy instructions to form translated instructions,

6 a translation store for storing the translated instructions,

7 an execution unit for executing said translated instructions to emulate said legacy
8 instructions,

9 an index table for storing translation indications for indicating translated blocks at block
10 numbers determined by said block addresses, said index table storing translation
11 indications using a subset of block address digits whereby block numbers in said
12 table are the same for multiple different blocks,

13 where for each of the legacy instruction of a translated block having a block number in said
14 table, said translation store includes one or more translated instructions for emulating
15 the legacy instruction, and,

16 if the legacy instruction is not a store instruction, the computer system goes
17 to the execution unit for executing said translated instructions,

18 if the legacy instruction is a store instruction, where the store instruction
19 stores to a particular block with a particular block number in said
20 table, the computer system checks the indications in said table for
21 said particular block number and,

CLEAN CLAIMS AFTER *RESPONSE A*

22 if the indications indicate that said particular block has not
23 been translated, the computer system goes to the
24 execution unit for executing said translated
25 instructions,

26 if the indications indicate that said particular block has not
27 been translated, said translation store is checked to
28 determine if instruction data has been modified and,
29 if modified, the translator repeats translating the
30 legacy instructions and the computer system goes to
31 the execution unit for executing said translated
32 instructions, and otherwise, if instruction data has not
33 been modified, the computer system goes to the
34 execution unit for executing said translated
35 instructions.

1 16. (Original) The system of Claim 15 wherein said index table stores indications for only a subset
2 of all the translated blocks.

1 17. (Original) The system of Claim 16 including a cache and wherein said subset of all the
2 translated blocks is stored in said cache.

1 18. (Cancelled).

1 19. (First Amended per A) The system of Claim 15 wherein said block address digits are included
2 in a three digit hexadecimal address field and said subset of block address digits is the center digit.

1 20. (Original) The system of Claim 15 wherein said legacy instructions are for a legacy system
2 having a S/390 architecture.

CLEAN CLAIMS AFTER *RESPONSE A*

1 21. (Original) The system of Claim 15 wherein said legacy instructions are object code instructions
2 compiled/assembled for a legacy architecture.

1 22. (Original) The system of Claim 15 wherein said legacy instructions include store instructions
2 for modifying instruction code.

1 23. (Original) The system of Claim 15 wherein said index table includes a state field for each block
2 number indicating whether the block represented by said block number has been modified.

1 24. (First Amended per A) The system of Claim 15 wherein,
2 said system includes a cache for storing said subset of all the translated blocks,
3 said index table includes a state field storing a count for each block number indicating
4 whether the block represented by said block number has been modified,
5 said count in a state field is incremented each time a block represented by said block number
6 has been modified in said cache,
7 said count in a state field is decremented each time a block represented by said block number
8 has been removed from said cache,
9 said translation store is not checked only when said count is zero.

CLEAN CLAIMS AFTER *RESPONSE A*

1 25. (First Amended per A) A computer system for dynamic emulation of object code legacy
2 instructions, where the legacy instructions have instruction addresses determined by
3 compilation/assembly of source code and where the legacy instructions include self-modifying store
4 instructions for modifying instruction code, comprising:

5 a group access unit for accessing blocks of said legacy instructions, said blocks having block
6 addresses,

7 a translation store for storing translation information for each of the legacy instructions,

8 an index table for storing translation indications, for only a subset of all the translated blocks

9 at block numbers determined by said block addresses, said index table storing

10 translation indications using a subset of block address digits whereby block numbers

11 in said table are the same for multiple different blocks and including a state field

12 storing a count for each block number indicating whether the block represented by

13 said block number has been modified by self-modifying store instructions,

14 a cache for storing translated instructions,

15 an execution unit for executing said translated instructions to emulate said legacy
16 instructions,

17 a legacy code translator operating, for each of the legacy instructions of said subset of all the

18 translated blocks having a block number in said table, to translate the legacy

19 instructions into one or more translated instructions for emulating the legacy

20 instruction and to store said translated instructions in the cache and,

21 if the legacy instruction is not a store instruction, the computer system goes

22 to said execution unit for executing said translated instructions,

23 if the legacy instruction is a store instruction, where the store instruction

24 stores to a particular block with a particular block number in said

25 table, the computer system checks the indications in said table for

26 said particular block number and,

27 if the indications indicate that said particular block has not

28 been translated, the computer system goes to said

CLEAN CLAIMS AFTER *RESPONSE A*

29 execution unit for executing said translated
30 instructions,
31 if the indications indicate that said particular block has been
32 translated, the computer system checks to determine
33 if instruction data has been modified and if modified,
34 the computer system goes to said translator to
35 translate the legacy instructions into one or more
36 translated instructions and the computer system goes
37 to said execution unit for executing said translated
38 instructions; and otherwise, if instruction data has not
39 been modified, the computer system goes to said
40 execution unit for executing said translated
41 instructions.

1 26. (First Amended per A) The system of Claim 25 wherein said count in a state field is
2 incremented each time a block represented by said block number has been modified in said cache,
3 said count in a state field is decremented each time a block represented by said block number has
4 been removed from said cache, said bypassing said repeating occurring only when said count is zero.

1 27. (Original) The system of Claim 25 wherein said legacy code is compiled/assembled for a native
2 architecture and executes as a guest on a host architecture.

1 28. (Original) The system of Claim 27 wherein the native architecture employs CISC instructions
2 and the host architecture employs RISC instructions.